1. USER 테이블 생성

CREATE TABLE USER\_LIST(

USER\_ID varchar2(20)not null primary key ,

USER\_PASSWORD varchar2(20) not null,

USER\_NAME varchar2(10) not null,

USER\_EMAIL varchar2(50) not null,

USER\_PHONE varchar2(15) not null,

USER\_DATE date

USER\_VISIBLE varchar2(1)

);

-----------------------------------------------------------------------------------

\* 회원가입 INSERT QUERY \* (확인 OK)

<insert id="insertUser\_List" parameter="hashMap">

<![CDATA[

INSERT INTO USER\_LIST(

USER\_ID,

USER\_PASSWORD,

USER\_NAME,

USER\_EMAIL,

USER\_PHONE,

USER\_DATE,

USER\_VISIBLE

) VALUES (

#{user\_id},

#{user\_password},

#{user\_name},

#{user\_email},

#{user\_phone},

SYSDATE,

'N'

)

]]>

</insert>

\* 아이디 중복 확인 QUERY \* (확인 ok)

<select id="selectIdCheck" parameterType="hashMap" resultType="Integer">

<![CDATA[

select count(\*) from USER\_LIST where USER\_ID=#{user\_id}

]]>

</select>

\* 이메일 인증 시 n->y로 바꿔주는 \* (확인 ok)

<update id="updateVerify" parameterType="hashmap">

<![CDATA[

update USER\_LIST set USER\_VISIBLE='Y' where USER\_EMAIL=#{user\_email}

]]>

</update>

\* 로그인 \* (ok)

<select id="selectLogin" parameterType="hashMap" resultType="hashMap">

<![CDATA[

select USER\_ID from USER\_LIST where USER\_ID=#{user\_id} and USER\_PASSWORD=#{user\_password}

]]>

</select>

\* 아이디 찾기 \* (확인ok, 이름과 이메일 입력으로 확인(VISIBLE가 Y 확인으로 이메일 인증까지 완료되서

활성화된 아이디만 검색))

<select id="selectId" parameterType="hashmap" resultType="hashmap">

<![CDATA[

SELECT

USER\_ID,

USER\_NAME

FROM

USER\_LIST

WHERE

USER\_NAME = #{user\_name} and USER\_EMAIL = #{user\_email} and

USER\_VISIBLE ='Y'

]]>

</select>

\* 비밀번호 찾기 \* (확인 ok, 아이디와 이메일 확인(VISIBLE가 Y 확인으로 이메일 인증까지 완료되서

활성화된 아이디만 검색))

<select id="selectId" parameterType="hashmap" resultType="hashmap">

<![CDATA[

SELECT

USER\_ID,

USER\_NAME,

USER\_PASSWORD

FROM

USER\_LIST

WHERE

USER\_ID = #{user\_id} and USER\_EMAIL = #{user\_email} and

USER\_VISIBLE ='Y'

]]>

</select>

\* 마이 페이지 들어가기 전 패스워드 확인 \* (보류 (쿼리는 맞지만 이방식으로 할지 고민))

<select id="selectMyLogin" parameterType="hashMap" resultType="hashMap">

<![CDATA[

SELECT

USER\_ID

FROM

USER\_LIST

WHERE

USER\_PASSWORD = #{user\_password}

]]>

</select>

\* 마이 페이지 내 정보 보기 \* (확인 ok)

<select id="selctMyInfo" parameterType="hashmap" resultType="hashmap">

<![CDATA[

SELECT

USER\_ID,

USER\_PASSWORD,

USER\_NAME,

USER\_EMAIL,

USER\_PHONE,

USER\_DATE

FROM

USER\_LIST

WHERE

USER\_ID = #{user\_id}

]]>

</select>

\* 내 정보 수정 \* (확인ok, 패스워드 변경까지)

<update id="updateMyInfo" parameterType="hashmap" >

<![CDATA[

UPDATE

USER\_LIST

SET

USER\_PASSWORD = #{user\_password},

USER\_EMAIL = #{user\_email},

USER\_PHONE = #{user\_phone}

WHERE USER\_ID = #{user\_id}

]]>

</update>

\* 회원탈퇴 \*

<delete id="deleteUser\_Id" parameterType="hashmap">

<![CDATA[

delete from USER\_LIST where USER\_ID=#{user\_id}

]]>

</delete>

-----------------------------------------------------------------------------------

\* 회원가입 INSERT QUERY \*(이거 아님XXXXXXXXXXX)

INSERT INTO USER\_LIST(

USER\_ID, USER\_PASSWORD, USER\_NAME, USER\_EMAIL, USER\_PHONE, USER\_DATE

)

values(

#USER\_ID#, #USER\_PASSWORD#, #USER\_NAME#, #USER\_EMAIL#, #USER\_PHONE#, sysdate

);

EX>

INSERT INTO USER\_LIST(

USER\_ID, USER\_PASSWORD, USER\_NAME, USER\_EMAIL, USER\_PHONE, USER\_DATE

)

values(

'HONG', '1234', '홍길동', 'HONG@NAVER.COM', '010-1234-1234', sysdate

);

------------------------------------------------------------------------------------

2 공지사항 테이블 생성

CREATE TABLE ADMIN\_NOTICE\_LIST(

NTC\_NUM NUMBER NOT NULL PRIMARY KEY,

NTC\_TITLE VARCHAR2(100) NOT NULL,

NTC\_CONTENT VARCHAR2(2000) NOT NULL,

NTC\_DATE DATE DEFAULT SYSDATE NOT NULL,

NTC\_HIT NUMBER DEFAULT 0

);

CREATE SEQUENCE NTC\_NUM\_SEQ

START WITH 1

INCREMENT BY 1

NOMAXVALUE

NOCACHE;

공지사항 리스트

<select id="selectNoticeList" parameterType="hashMap" resultType="hashMap">

<![CDATA[

SELECT

NTC\_NUM,

NTC\_TITLE,

NTC\_DATE

FROM

ADMIN\_NOTICE\_LIST

order by NTC\_NUM desc

]]>

</select>

\* 관리자 공지사항 등록 QUERY \* (확인 ok)

<insert id="insertNoticeWrite" parameterType="hashMap">

<![CDATA[

INSERT INTO

ADMIN\_NOTICE\_LIST (

NTC\_NUM,

NTC\_TITLE,

NTC\_CONTENT,

NTC\_DATE,

NTC\_HIT

) VALUES (

NTC\_NUM\_SEQ.NEXTVAL,

#{NTC\_TITLE},

#{NTC\_CONTENT},

SYSDATE,

'0'

)

]]>

</insert>

\* 관리자 공지사항 수정 QUERY \* (확인 OK)

<update id="updateNotice" parameterType="hashMap">

<![CDATA[

UPDATE ADMIN\_NOTICE\_LIST

SET

NTC\_TITLE = #{NTC\_TITLE},

NTC\_CONTENT = #{NTC\_CONTENT},

NTC\_DATE = SYSDATE

WHERE

NTC\_NUM = #{num}

]]>

</update>

\* 관리자 공지사항 상세보기 \* (확인 OK, 카운트 올라가는지 확인해야함)

<select id="selectNoticeDetail" resultType="hashMap" parameterType="hashMap">

<![CDATA[

SELECT

NTC\_NUM,

NTC\_TITLE,

NTC\_CONTENT,

NTC\_DATE,

NTC\_HIT

FROM

ADMIN\_NOTICE\_LIST

WHERE

NTC\_NUM = #{num}

]]>

</select>

\* 글 삭제 \* (확인 OK)

<delete id="deleteNotice" parameterType="hashmap">

<![CDATA[

DELETE FROM ADMIN\_NOTICE\_LIST WHERE NTC\_NUM = #{num}

]]>

</delete>

-------------------------------------------------------------------------------------

3. 1:1 문의 게시판 테이블 생성

CREATE TABLE ADMIN\_QNA\_LIST(

USER\_ID VARCHAR2(20) NOT NULL,

ADQNA\_NUM NUMBER NOT NULL PRIMARY KEY,

ADQNA\_TITLE VARCHAR2(100) NULL,

ADQNA\_CNT VARCHAR2(2000),

ADQNA\_CMT VARCHAR2(2000),

QA\_ANS VARCHAR2(1) NOT NULL,

ADQNA\_DATE DATE DEFAULT SYSDATE NOT NULL,

FOREIGN KEY (USER\_ID) REFERENCES USER\_LIST (USER\_ID)

);

CREATE SEQUENCE ADQNA\_NUM\_SEQ

START WITH 1

INCREMENT BY 1

NOMAXVALUE

NOCACHE;

1:1 문의 게시판 등록 (사용자 질문) (확인ok)

<insert id="insertUserAdQNAWrite" parameterType="hashMap" >

<![CDATA[

INSERT INTO

ADMIN\_QNA\_LIST (

USER\_ID,

ADQNA\_NUM,

ADQNA\_TITLE,

ADQNA\_CNT,

QA\_ANS,

ADQNA\_DATE

) VALUES (

#{user\_id},

ADQNA\_NUM\_SEQ.NEXTVAL,

#{ADQNA\_TITLE},

#{ADQNA\_CNT},

'N',

SYSDATE

)

]]>

</insert>

\* 1:1 문의 게시판 리스트 (관리자 화면) (확인ok)

<select id="selectAdQNAList" parameterType="hashMap" resultType="hashMap">

<![CDATA[

SELECT

USER\_ID,

ADQNA\_NUM,

ADQNA\_TITLE,

ADQNA\_DATE,

QA\_ANS

FROM

ADMIN\_QNA\_LIST

order by ADQNA\_NUM desc

]]>

</select>

\* 1:1 문의 게시판 리스트 (사용자 화면) (user\_id 에 사용자 아이디 것만의 정렬로불러오기)

<select id="selecUserAdQNAList" parameterType="hashMap" resultType="hashMap">

<![CDATA[

SELECT

ADQNA\_NUM,

ADQNA\_TITLE,

ADQNA\_DATE,

QA\_ANS

FROM

ADMIN\_QNA\_LIST

WHERE

USER\_ID = #{user\_id}

order by ADQNA\_NUM desc

]]>

</select>

1:1 문의 게시판 상세보기 (확인OK)

<select id="selectAdQNADetail" resultType="hashMap" parameterType="hashMap" >

<![CDATA[

SELECT

USER\_ID,

ADQNA\_NUM,

ADQNA\_TITLE,

ADQNA\_CNT,

ADQNA\_CMT,

QA\_ANS,

ADQNA\_DATE

FROM

ADMIN\_QNA\_LIST

WHERE

ADQNA\_NUM = #{adqna\_num}

]]>

</select>

1:1 문의 게시판 답변하기(관리자) (뭔가이상해서 수정필요)

<insert id="insertAdQNAWrite" parameterType="hashMap" >

<![CDATA[

INSERT INTO

ADMIN\_QNA\_LIST (

ADQNA\_NUM,

ADQNA\_TITLE,

ADQNA\_CMT,

QA\_ANS,

ADQNA\_DATE

) VALUES (

ADQNA\_NUM\_SEQ.NEXTVAL,

#{NT\_TITLE},

#{NT\_CMT},

'Y',

SYSDATE

)

]]>

</insert>

후 질문자의 답변등록 하면 질문자 글의 QA\_ANS가 'N'에서 'Y'로 연이어 바꿈 실행

<update id="updateAdAns" parameterType="hashmap">

<![CDATA[

update ADMIN\_QNA\_LIST set QA\_ANS ='Y' where ADQNA\_NUM=#{adqna\_num}

]]>

</update>

1:1 질문 수정

<update id="updateAdQNA" parameterType="hashMap">

<![CDATA[

UPDATE ADMIN\_QNA\_LIST

SET

ADQNA\_TITLE = #{adqna\_title},

ADQNA\_CNT = #{adqna\_cnt}

WHERE

ADQNA\_NUM = #{num}

]]>

</update>

-----------------------------------------------------------------------------------------------------------

4. 공간 신청 테이블 생성 (안됨 확인할것)

CREATE TABLE SPACE\_APLLY\_LIST(

APPLY\_NUM NOT NULL PRIMARY KEY,

APPLY\_TITLE VARCHAR2(100) NOT NULL,

APPLY\_DETAIL VARCHAR2(2000),

APPLY\_CNT VARCHAR2(2000),

APPLY\_USE VARCHAR2(100) NOT NULL,

APPLY\_IMG VARCHAR2(1000) NOT NULL,

APPLY\_POS VARCHAR2(1000) NOT NULL,

APPLY\_PRI NUMBER NOT NULL,

APPLY\_SIZE NUMBER NOT NULL,

APPLY\_FIOOR NUMBER NOT NULL,

SPACE\_TAG VARCHAR2(500),

APPLY\_OPT VARCHAR2(1000) NOT NULL,

APPLY\_ID VARCHAR2(20) NOT NULL,

APPLY\_MODIFY VARCHAR2(1) DEFAULT 'N' NOT NULL,

FOREIGN KEY (APPLY\_ID) REFERENCES USER\_LIST (USER\_ID)

);

----------------------------------------------------------------------------------------------------

....... 찜 목록 테이블 생성 (FOREING 부모키는 PRIMARY KEY 여야 함)(아직 테스트가 안됨XXXX)

CREATE TABLE LIKE\_LIST(

USER\_ID VARCHAR2(20) NOT NULL,

SPACE\_ID NUMBER NOT NULL,

FOREIGN KEY (USER\_ID) REFERENCES USER\_LIST (USER\_ID),

FOREIGN KEY (SPACE\_ID) REFERENCES SPACE\_LIST (SPACE\_ID)

);

........... 찜하기

............찜목록 불러오기

<select id="selecResList" parameterType="hashMap" resultType="hashMap">

<![CDATA[

SELECT

ADQNA\_NUM,

ADQNA\_TITLE,

ADQNA\_DATE,

QA\_ANS

FROM

ADMIN\_QNA\_LIST

WHERE

USER\_ID = #{user\_id}

order by ADQNA\_NUM desc

]]>

</select>

-------------------------------------------------------찜목록 불러오기는 아직 작성 안한것

----------------------------------------------------------------------------------------------------

.... 후기 게시판 테이블 생성 (공간 리스트를 먼저 만들어야함)

CREATE TABLE REPLY\_LIST(

SPACE\_ID NUMBER NOT NULL,

USER\_ID VARCHAR2(20) NOT NULL,

REPLY\_NUM NUMBER NOT NULL PRIMARY KEY,

REPLY\_CNT VARCHAR2(1000) NOT NULL,

REPLY\_CMT VARCHAR2(1000) NOT NULL,

REPLY\_IMG VARCHAR2(2000),

REPLY\_DATE DATE DEFAULT SYSDATE,

REPLY\_DEL VARCHAR2(1) DEFAULT 'N' NOT NULL,

FOREIGN KEY(SPACE\_ID) REFERENCES SPACE\_LIST(SPACE\_ID),

FOREIGN KEY(USER\_ID) REFERENCES USER\_LIST(USER\_ID)

)

CREATE SEQUENCE REPLY\_NUM\_SEQ

START WITH 1

INCREMENT BY 1

NOMAXVALUE

NOCACHE;

---------------------------------------------------------------------------------------------------

....... 공간 문의 게시판 (공간 리스트를 먼저 만들어야함)

CREATE TABLE REPLY\_LIST(

SPACE\_ID NUMBER NOT NULL,

USER\_ID VARCHAR2(20) NOT NULL,

QNA\_NUM NUMBER NOT NULL PRIMARY KEY,

QNA\_TITLE VARCHAR2(100) NOT NULL,

QNA\_CNT VARCHAR2(1000) NOT NULL,

QNA\_DATE DATE DEFAULT SYSDATE,

QNA\_ANS VARCHAR2(1) DEFAULT 'N' NOT NULL,

QNA\_DEL VARCHAR2(1) DEFAULT 'N' NOT NULL,

FOREIGN KEY(SPACE\_ID) REFERENCES SPACE\_LIST(SPACE\_ID),

FOREIGN KEY(USER\_ID) REFERENCES USER\_LIST(USER\_ID)

)

CREATE SEQUENCE QNA\_NUM\_SEQ

START WITH 1

INCREMENT BY 1

NOMAXVALUE

NOCACHE;

---------------------------------------------------------------------------------------------------

.......